HAL open science

Fixed-sequence single machine scheduling and outbound delivery problems

Azeddine Cheref, Alessandro Agnetis, Christian Artigues, Jean-Charles Billaut

► To cite this version:

Azeddine Cheref, Alessandro Agnetis, Christian Artigues, Jean-Charles Billaut. Fixed-sequence single machine scheduling and outbound delivery problems. 5th International Conference in Operations Research and Enterprise systems (ICORES'2016), Feb 2016, Rome, Italy. hal-01304785

HAL Id: hal-01304785
https://hal.science/hal-01304785

Submitted on 26 Aug 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fixed-sequence single machine scheduling and outbound delivery problems

Azeddine Cheref[1,2], Alessandro Agnetis[3], Christian Artigues[2], Jean-Charles Billaut[1]
[1] *Université François-Rabelais Tours / CNRS, 64 av. J. Portalis, 37200 Tours, France*
[2] *LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France*
[3] *Dipartimento di Ingegneria dell'Informazione, Universitá degli Studi di Siena, via Roma 56, 53100 Siena, Italy*

Keywords: Scheduling; Batching and delivery; Computational complexity; Dynamic programming

Abstract: In this paper, we study an integrated production an outbound delivery scheduling problem with a predefined sequence. The manufacturer has to process a set of jobs on a single machine and deliver them in batches to multiple customers. A single vehicle with limited capacity is used for the delivery. Each job has a processing time and a specific customer location. Starting from the manufacturer location, the vehicle delivers a set of jobs which constitute a batch by taking into account the transportation times. Since the production sequence and delivery sequence are fixed and identical, the problem consists in deciding the composition of batches. We prove that for any regular sum-type objective function of the delivery times, the problem in NP-hard in the ordinary sense and can be solved in pseudopolynomial time. A dynamic programming algorithm is proposed.

1 INTRODUCTION

This paper deals with a model for coordinating production and delivery schedules. In many production systems, finished products are delivered from the factory to multiple customer locations, warehouses, or distribution centers by delivery vehicles. An increasing amount of research has been devoted, during the last twenty years, to devise integrated models for production and distribution. These models have been largely analyzed and reviewed by (Chen, 2010), who proposed a detailed classification scheme. The models reflect the variety of issues, including systems structure, vehicle/transportation system characteristics, delivery modes, various types of time constraints. In the large majority of the models in the literature, the coordination of production and distribution is achieved through the creation of *batches*, i.e., several parts are shipped together and delivered to their respective destinations during a single trip. When forming batches, one must therefore take into account both production information (such as processing time, release dates etc) and delivery information (such as customer location, time windows etc). Most of the models presented in the literature explicitly take into account transportation times to reach the customers' location, but there are no proper routing decisions, since the number of distinct customers is typically very small. Hence, the focus of the analysis is often on scheduling and batching.

Many models consider delivery as a separate step after production, but do not model it in details, e.g. assuming that a sufficiently large number of vehicles is available to deliver the products at any time (Chen and Lee, 2008) (Agnetis et al., 2014) (Fan et al., 2015), or assuming that there is a single customer. In (Chang and Lee, 2004) the jobs have a certain size and the capacity of a vehicle is a physical space available, and jobs have to be delivered to a unique customer. NP-hardness results are given as well as heuristic algorithms with performance guarantee. In (Li and Ou, 2005) and (Wang and Cheng, 2009), delivery concerns the materials as well as finished jobs which must be transported to a single warehouse. The objective is to minimize the delivery time of the job delivered last to this customer. Li and Ou propose a polynomial time algorithm when the production sequence is fixed. Delivery can also be modeled as a delay after the production completion time. Fan, Lu and Liu (Fan et al., 2015) consider the join problem of scheduling and routing with availability constraints of the machine. The delivery is performed by an unlimited number of uncapacitated vehicles and the objective is to minimize the total delivery time and total delivery cost. Depending on the problem they consider, the authors propose polynomial time algorithms, an algorithm with guaranteed performance and a polynomial time approximation scheme.

In the paper by (Li et al., 2005), a single vehicle is used for delivery, and hence the vehicle schedule has to be synchronized with production and batching decisions. In particular, one must also take account of the time that the vehicle will take to deliver a certain batch of products, and that it will take to be back at the manufacturing facility. (Li et al., 2005) analyze the joint problem of production sequencing and batch formation, in order to minimize total delivery time, given that delivery is performed by a single vehicle. Total delivery time is a meaningful indicator of the overall efficiency of the delivery process. They show that in general the problem is NP-hard, and then give polynomial time algorithms for the problem with a fixed number of distinct destinations. In (Tsirimpas et al., 2008), the authors consider that all the jobs are ready for the delivery at the beginning of the time horizon (no scheduling problem here). The delivery is performed by a single capacitated vehicle and the sequence of delivery is predefined. The objective is to minimize the total travel time and the authors propose polynomial time algorithms. In (Chen and Lee, 2008), the authors consider the problem with a single machine where finished jobs must be transported by an unlimited number of vehicles. There is no vehicle routing problem, since the vehicle delivers in one trip only jobs delivered at the same destination.

Using the terminology of (Chen, 2010), the models presented in this paper concern *batch delivery with routing*, i.e., orders going to different customers can be delivered together in the same shipment (batch).

A distinctive feature of the problems we address here is that the jobs must be delivered in the same order in which they are produced. Examples of situations in which the customer sequence is *fixed* are reported by (Armstrong et al., 2008), (Viergutz and Knust, 2014) and include a fixed sequence of customers and a single round trip for the delivery (hence, the production sequence and the routing sequence are the same), as well as the objective is to maximize the total demand without violating the product lifespan. This problem is proved NP-hard by (Armstrong et al., 2008). In (Lenté and Kergosien, 2014), the authors consider that the production sequence is fixed as well as the delivery sequence. The authors search for a batching of jobs minimizing the makespan, the maximum lateness and the number of tardy jobs. The problems are modeled by a graph and polynomial time algorithms are proposed for these objective functions.

In this paper, we will mainly focus on the problem of deciding how to form batches with a given production sequence (problem P1). We completely characterize the complexity of Problem P1, showing that when the objective function is to minimize the total delivery time it is NP-hard in the ordinary sense, and that it can be solved in pseudopolynomial time for any sum-type function of the delivery times.

The paper is organized as follows. In Section 2 we present the problem formally and show that the problem is NP-hard when Z is the total delivery time, and that it can be solved in pseudopolynomial time when Z is any sum-type objective function. Finally, some conclusions and future research directions are presented in Section 4.

2 PROBLEM DEFINITION AND COMPLEXITY

2.1 Problem definition and notation

The problem considered in this paper can be described as follows. A set of n jobs is given and their production sequence is known. Each job J_j, $j = 1,\dots,n$, requires a certain *processing time* p_j on a single machine, and must be delivered to a certain location site. For the sake of simplicity, when it does not create confusion, we use j to refer to the destination of job J_j. We denote by $t_{i,j}$ the transportation time from destination i to destination j. For analogy with vehicle routing problems, we refer to the manufacturer's location as the *depot*. We use M to denote the depot (manufacturer), hence $t_{M,j} = t_{j,M}$ is the transportation time between the depot and destination j. Unless otherwise specified, we assume that transportation times are symmetric and satisfy the triangle inequality.

Deliveries are carried out by a single *vehicle*. The vehicle loads a certain number of jobs which have been processed and departs towards the corresponding destinations. Thereafter, it returns to the depot, hence completing a *round trip*. The set of jobs delivered during a single round trip constitutes a *batch*. The capacity c of the vehicle is the maximum number of jobs it can load and hence deliver in a round trip. The jobs must be delivered in the order in which they are produced, hence the production sequence also specifies the sequence in which the customers have to be reached.

The problem consists in deciding a partition of all jobs into *batches*, i.e., a *batching scheme*. Each batch will be routed according to the manufacturing sequence. In general, the performance of the system depends on all the concurrent decisions: production scheduling, batching and vehicle routing. This requires therefore an *integrated model*, allowing one to coordinate all these aspects. A *solution* to our problem with fixed sequence, is the specification of a batching scheme. Given a solution, we denote by C_j the *completion time* of job J_j on the single machine, which is also the time at which the job is released for delivery, i.e., the batch including job J_j cannot start before C_j. We denote by D_j the delivery time of J_j, i.e., the time at which the job J_j is delivered at its destination. The performance measures we consider in this paper depend on such delivery times. In particular, denoting with Z the performance measure, in this paper we consider:

- the *total delivery time*, i.e., $Z = \sum_{j=1}^{n} D_j$
- a *general sum-type performance index*, i.e., $Z = \sum_{j=1}^{n} f_j(D_j)$, where $f_j(D_j)$ is a general, nondecreasing function of D_j, $j = 1,\ldots,n$.

Note that the latter case includes total (weighted) delivery time, total (weighted) tardiness, etc.

We consider the following problem:

Problem $P1(Z)$**.** *Given n jobs of length p_j, $j = 1,\ldots,n$, transportation times $t_{i,j}$ for all i, j, and a sequence σ, find a batching scheme $\mathcal{B}$ such that Z is minimized.*

2.2 Complexity

Since the production sequence is given, and since jobs are delivered to the respective customers in the same given order, we assume that the job sequence is $\sigma = (J_1, J_2, ..., J_n)$. Only travel times $t_{j,j+1}$ are relevant, as well as times $t_{j,M} = t_{M,j}$, representing the travel time between customer j and the manufacturer or vice-versa.

Let us first consider the problem in which the objective function is the total delivery time, i.e., problem $P1(\sum_{j=1}^{n} D_j)$. For our purposes, we introduce the following problem.

EVEN-ODD PARTITION (EOP). A set of n pairs of positive integers $(a_1,b_1),(a_2,b_2),\ldots,(a_n,b_n)$ is given, in which, for each i, $a_i > b_i$. Letting $K = \sum_{i=1}^{n}(a_i + b_i)$, is there a partition $(S,\bar{S})$ of the index set $I = \{1,2,\ldots,n\}$ such that

$$\sum_{i\in S} a_i + \sum_{i\in\bar{S}} b_i = K/2? \tag{1}$$

EOP is NP-hard in the ordinary sense (Garey et al., 1988). In the following, we will actually use the following slightly modified version of the problem.

MODIFIED EVEN-ODD PARTITION (MEOP). A set of n pairs of positive integers $(a_1,b_1),(a_2,b_2)$, $\ldots,(a_n,b_n)$ is given, in which, for each i, $a_i > b_i$. Letting $Q = \sum_{i=1}^{n}(a_i - b_i)$, is there a partition $(S,\bar{S})$ of the index set $I = \{1,2,\ldots,n\}$ such that

$$\sum_{i\in S}(a_i - b_i) = Q/2? \tag{2}$$

Note that the two problems are indeed equivalent. In fact, suppose that EOP has a partition $(S,\bar{S})$. The corresponding instance of MEOP also admits the same partition. In fact, subtracting $\sum_{i=1}^{n} b_i = \sum_{i\in S} b_i + \sum_{i\in\bar{S}} b_i$ from both sides of (1), one obtains

$$\sum_{i\in S}(a_i - b_i) = K/2 - \sum_{i=1}^{n} b_i \tag{3}$$

Now, from the definitions of K and Q it turns out that

$$Q = K - 2\sum_{i=1}^{n} b_i$$

and hence (3) is indeed (2). We next show the following result.

Theorem 2.1. *$P1(\sum_{j=1}^{n} D_j)$ is NP-hard.*

Proof. The problem is obviously in NP. Given an instance of MEOP, we build an instance of P1 as follows. There are $3n+3$ jobs. The processing times of the jobs are defined as follows:

- $p_1 = p_2 = p_3 = 0$.
- for each $i = 0,1,\ldots,n-1$, one has
  - $p_{3i+1} = p_{3i+2} = 1$,
  - $p_{3i+3} = 4x_i + b_i - 2$.
- $p_{3n+1} = 4x_n + b_n + Q/2$.
- $p_{3n+2} = p_{3n+3} = 0$,

  where the x_i are defined by

$$x_i = (3a_i - 2b_i + 3(n-i)(a_i - b_i))/2 \ \forall\, i = 1\ldots n \tag{4}$$

and $x_{n+1} = 0$.

In the following, we refer to the set of jobs $(J_{3i+1}, J_{3i+2}, J_{3i+3})$, $i = 0,\ldots,n$, as the *triple* T_{i+1}.

For what concerns the travel times, we let:

- for each $i = 0,1,\ldots,n-1$, one has
  - $t_{M,3i+1} = t_{3i+1,M} = t_{M,3i+2} = t_{3i+2,M} = t_{M,3i+3} = t_{3i+3,M} = x_{i+1}$,
  - $t_{3i+1,3i+2} = a_{i+1}$,
  - $t_{3i+2,3i+3} = b_{i+1}$,
  - $t_{3i+3,3i+4} = x_{i+1} + x_{i+2}$.
- $t_{M,3n+1} = t_{3n+1,M} = t_{M,3n+2} = t_{3n+2,M} = t_{M,3n+3} = 0$.
- $t_{3n+1,3n+2} = t_{3n+2,3n+3} = 0$.

Finally, vehicle capacity is $c = 2$. The problem consists in determining whether a solution exists such that the total delivery time does not exceed

$$f^* = \sum_{i=1}^{n}(3C_{3i} + 7x_i + b_i) + C_{3n+1} + C_{3n+2} + C_{3n+3} - Q/2. \tag{5}$$

For shortness, we call *feasible* a schedule satisfying (5). The proof has the following scheme.

1. We first establish that if a feasible schedule exists, then there is one having a certain structure, called *triple-oriented*,
2. We analyze some properties of this structure,
3. We show that a triple-oriented schedule of value f^* exists if and only if EOP is a yes-instance.

Lemma 2.2. *If a feasible schedule exists, then there exists one satisfying the following property: for all $i = 1, \dots, n+1$, jobs J_{3i} and J_{3i+1} are NOT in the same batch.*

Proof. Suppose that a feasible schedule exists in which, for a certain i ($1 \le i \le n$), jobs J_{3i} and J_{3i+1} are in the same batch. Since $c = 2$, the batch contains no other job. As a consequence, after delivering J_{3i+1}, the vehicle must go back to M in order to load the next jobs and start a new trip. If we denote by τ the start time of the round trip of jobs J_{3i} and J_{3i+1}, job J_{3i} is delivered at time $D_{3i} = \tau + t_{M,3i}$ and job J_{3i+1} is delivered at time $D_{3i+1} = \tau + t_{M,3i} + t_{3i,3i+1}$. Therefore we have $D_{3i} = \tau + x_i$ and $D_{3i+1} = \tau + x_i + (x_i + x_{i+1})$. The vehicle is back at M at time $\tau + 2x_i + 2x_{i+1}$. Now, if we replace this batch with two batches of one job each, the delivery times of both jobs as well as the time at which the vehicle is back at M are unchanged. Therefore, there is an equivalent solution where J_{3i} and J_{3i+1} are not in the same batch. □

We call *triple-oriented* a schedule satisfying Lemma 2.2. The reason of this name is that the schedule is decomposed according to triples. More precisely, since $c = 2$, for each triple $T_{i+1} = (J_{3i+1}, J_{3i+2}, J_{3i+3})$, $i = 0, \dots, n-1$, there are *exactly* two batches, and only two possibilities, namely:

- either the first batch is $\{J_{3i+1}, J_{3i+2}\}$ and the second is $\{J_{3i+3}\}$,
- or the first batch is $\{J_{3i+1}\}$ and the second is $\{J_{3i+2}, J_{3i+3}\}$.

We call these two possibilities *option A* and *option B* respectively (see Fig. 1). Namely, let us view option B as the *Base option*, and A as a variant to it.

Round trip length. Letting M_i^A and M_i^B denote the round trip length of the jobs of T_i in the two cases. One has:

$$M_i^A = 4x_i + a_i \tag{6}$$
$$M_i^B = 4x_i + b_i \tag{7}$$

Since $a_i > b_i$, option A implies a longer round trip length than the Base option. The difference between the two (i.e., the additional time with option A with respect to B) is precisely $a_i - b_i$.

Lemma 2.3. *In any triple-oriented schedule, the vehicle is never idle, except possibly before loading J_{3n+1}.*

Proof. Let consider the first triple T_1. The vehicle starts at time 0 (to deliver batch $\{J_1\}$ or $\{J_1, J_2\}$), and is back at time $4x_1 + b_1$ or at time $4x_1 + a_1$. The completion time of T_2 is precisely equal to $C_6 = \sum_{j=1}^{6} p_j = 1 + 1 + 4x_1 + b_1 - 2 = 4x_1 + b_1$. Therefore, the vehicle can immediately start the delivery of the jobs of T_2. For the same reasons, the delivery of the jobs of T_i cannot be smaller than, the duration of the jobs of T_{i+1}, and the vehicle will be able to start immediately the delivery of the jobs of T_{i+1}. This reasoning stops for the last triple T_{n+1} because the duration of J_{3n+1} is particular. □

In view of Lemma 2.3, one can compute the total delivery time in the Base scenario, i.e., when option B is *always* chosen. From (7), one has that the vehicle always returns to M exactly at time C_{3i}. Therefore, the last time the vehicle arrives at M (before delivering the jobs of T_{n+1}) is $C_{3n} + 4x_n + b_n$. Because of the definition of p_{3n+1}, and because J_{3n+1} starts at time C_{3n}, this time is equal to $C_{3n+3} - Q/2$. In this case, the vehicle will stay idle from $C_{3n+1} - Q/2$ to C_{3n+1}, when job J_{3n+1} can be finally loaded (see Fig.2) and delivered (the two reminder jobs have a duration of 0 and travel times equal to 0). Hence, we have:

$$f^{BASE} = \sum_{i=1}^{n} (3C_{3i} + 7x_i + b_i) + C_{3n+1} + C_{3n+2} + C_{3n+3} \tag{8}$$

Contribution to total delivery time. Before computing the contribution of a certain triple to the total delivery time, let us consider schedules in which the transportation of the last three jobs J_{3n+1}, J_{3n+2} and J_{3n+3} start exactly at their release time, i.e., at time $C_{3n+1} = C_{3n+2} = C_{3n+3}$ (options A and B are equivalent). Let us call *regular* a schedule in which such a condition holds.

Expression (8) refers to the scenario in which for all triples, the option B is chosen. We want now to compute the objective function of an arbitrary solution. Let us first consider the contribution of triple T_i to the objective function in the Base schedule, i.e., assuming that *the delivery of T_i started at time C_{3i}*, and let us denote such piece of contribution as TDT_i^A and TDT_i^B in the two cases. One has:

$$\begin{aligned} TDT_i^A &= (C_{3i} + x_i) + (C_{3i} + x_i + a_i) + (C_{3i} + 3x_i + a_i) \\ &= 3C_{3i} + 5x_i + 2a_i \end{aligned} \tag{9}$$

$$\begin{aligned} TDT_i^B &= (C_{3i} + x_i) + (C_{3i} + 3x_i) + (C_{3i} + 3x_i + b_i) \\ &= 3C_{3i} + 7x_i + b_i \end{aligned} \tag{10}$$

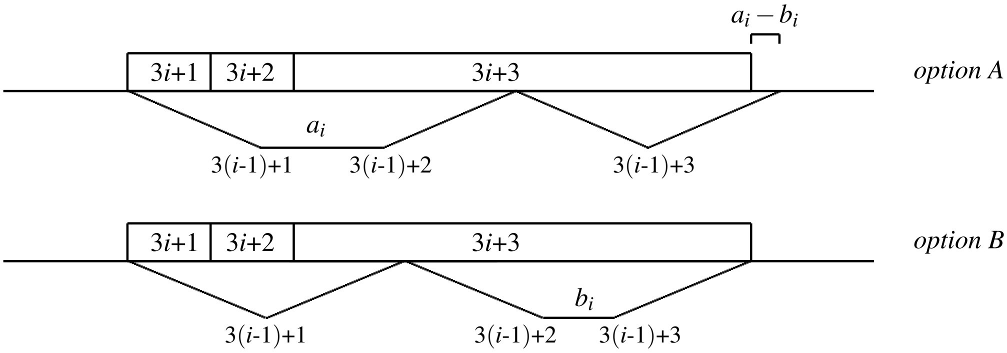

Figure 1: Round trips with options A and B.

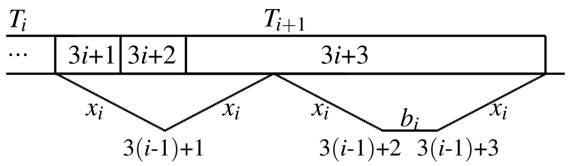

Figure 2: The base schedule (i.e., B is always chosen).

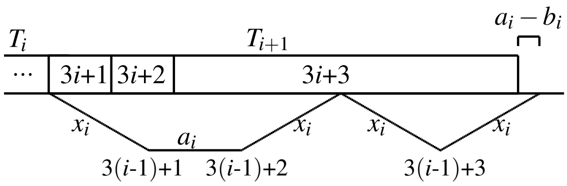

Figure 3: T_i is the first triple choosing option A.

Note that

$$
\begin{aligned}
TDT_i^B - TDT_i^A &= 2x_i + b_i - 2a_i \\
&= (3a_i - 2b_i + 3(n-i)(a_i - b_i)) + b_i - 2a_i \\
&= a_i - b_i + 3(n-i)(a_i - b_i) \qquad (11)
\end{aligned}
$$

which is positive, remembering that $a_i > b_i$. This means that choosing option A over B brings a benefit in terms of total delivery time. However, such favorable situation for option A is mitigated by the fact that, with option A, one has a longer round trip time than with option B, by the amount $(a_i - b_i)$ (Fig.3). In a regular schedule, such increased round trip time will be "paid" by all subsequent jobs, except the last jobs J_{3n+1}, J_{3n+2} and J_{3n+3}. Hence, in a regular schedule the total effect (in favor of option B) on the subsequent jobs of choosing option A for T_i is given by

$$3(n-i)(a_i - b_i) \qquad (12)$$

In conclusion, the *net benefit* of choosing option A over B for T_i in terms of objective function value is obtained subtracting (12) from (11), and in view of the definition of x_i (4), one has therefore that

$$
\begin{aligned}
NetBenefit_i &= (2x_i + b_i - 2a_i) - 3(n-i)(a_i - b_i) \\
&= a_i - b_i \qquad (13)
\end{aligned}
$$

In conclusion, it turns out that, when A is chosen over the Base option, one has a larger round trip time, by $(a_i - b_i)$, but also a smaller contribution to total delivery time (also by the amount $(a_i - b_i)$)(see Fig. 4). So, given any regular triple-oriented schedule in which the last three jobs depart at their completion time, let T_A be the set of triples for which the option A is chosen. Then, from the above considerations, the value f of the objective function is given by

$$f = f^{BASE} - \sum_{i \in T_A} (a_i - b_i) \qquad (14)$$

On the other hand, the time at which the vehicle returns to M before loading the last three jobs (J_{3n+1}, J_{3n+2} and J_{3n+3}) is given by

$$C_{3n} + 4x_n + b_n + \sum_{i \in T_A} (a_i - b_i) \qquad (15)$$

Now, in a regular schedule job J_{3n+1} (and also J_{3n+2} and J_{3n+3}) starts at time $C_{3n+1} = C_{3n} + 4x_n + b_n + Q/2$. Hence, from (15), in a regular schedule, it must hold:

$$\sum_{i \in T_A} (a_i - b_i) \leq Q/2$$

On the other hand, comparing (5), (8) it turns out that

$$f^* = f^{BASE} - Q/2$$

and hence, from (14), a regular schedule is feasible precisely if a subset T_A of indices exists such that $\sum_{i \in T} (a_i - b_i) = Q/2$, i.e., if and only if a feasible partition exists in the instance of EOP. To conclude the proof, it is left to show that f^* can be attained only by a regular schedule. In fact, if a schedule is not regular,

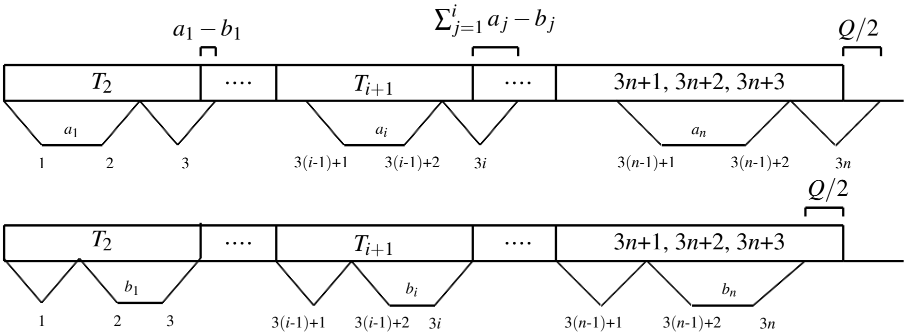

Figure 4: Round trips with option A only and option B only

the departure time of the last batch is delayed by the amount $(\sum_{i \in T_A}(a_i - b_i) - Q/2)$ with respect to C_{3n+1}. As a consequence, the expression of f in (14) must be modified to take account of such delay of the last three jobs, i.e. it comes

$$\begin{aligned} f &= f^{BASE} - \sum_{i \in T_A}(a_i - b_i) + 3(\sum_{i \in T_A}(a_i - b_i) - Q/2) \\ &= f^{BASE} + 2\sum_{i \in T_A}(a_i - b_i) - 3Q/2 \qquad (16) \end{aligned}$$

Since, in a nonregular schedule,

$$\sum_{i \in T_A}(a_i - b_i) > Q/2,$$

from (16) one has

$$f > f^{BASE} + Q - 3Q/2 = f^{BASE} - Q/2$$

and hence it cannot be feasible. □

3 PSEUDOPOLYNOMIAL TIME ALGORITHM FOR $P1(\sum_j f_j(D_j))$

Theorem 2.1 implies that no optimal polynomial time algorithm can be found for $P1(\sum_{j=1}^{n} D_j)$, and hence for more general objective functions, unless P=NP. In what follows, we show that $P1(\sum_j f_j(D_j))$ can be solved in pseudopolynomial time, hence settling the complexity status of P1.

In what follows we denote by $\{i, j\}$ the batch consisting of jobs $J_i, \ldots, J_j$. As usual, C_j is the completion time of job J_j (known because σ is known), and hence the release time for delivery. We denote by $M(i, j)$ the duration of the round trip of batch $\{i, j\}$, and, if the batch starts at time t, we call $K(i, j, t)$ its contribution to the objective function. Also, we assume that at the beginning, the vehicle is at the manufacturing location.

We denote by $F(i, j, t)$ the value of the optimal solution of the problem restricted to the first j jobs, in which the first job of the last batch is J_i, and such that the batch starts at time t. Then, $F(i, j, t)$ can be computed by means of a simple recursive formula. In the optimal solution of the subproblem, if the second last batch is $\{p, i-1\}$, and if it starts at time s, then we have:

$$F(i, j, t) = F(p, i-1, s) + K(i, j, t)$$

Note that, if the vehicle starts at time s, it must be back before or at time t, i.e., the following constraint must hold:

$$C_{i-1} \le s \le t - M(p, i-1)$$

In conclusion, the problem is solved by means of:

$$F(i, j, t) = \min_{\substack{\max(i-c,1) \le p \le i-1 \\ C_{i-1} \le s \le t - M(p,i-1)}} \{F(p, i-1, s)\} + K(i, j, t) \qquad (17)$$

Let T be an upper bound on the latest possible departure time for the last batch. As long as the triangle inequality holds, this is given, for instance, by:

$$T = \max\left(\max_{1 \le i \le n-1} \{C_i + 2\sum_{h=i}^{n-1} t_{hM}\}, C_n \right)$$

The optimal solution is given by:

$$z^* = \min_{n-c+1 \le i \le n, C_n \le t \le T} \left(F(i, n, t)\right) \qquad (18)$$

A few boundary conditions must be imposed:

$$\begin{aligned} F(i, j, t) &= +\infty \text{ for all } j < i \qquad (19) \\ F(1, j, t) &= K(1, j, t) \text{ for all } j, t \qquad (20) \end{aligned}$$

Condition (19) is obvious. Condition (20) allows to initialize the algorithm.

Let us turn to complexity. First, consider the computation of values $M(i, j)$ and $K(i, j, t)$. Both can be simply computed adding the contribution of the next job in the batch either to the round trip time (for $M(i, j)$) or to the objective function (for $K(i, j, t)$). More precisely, the delivery time d_h of job J_h *with respect to the departure time of the batch* is simply given by:

$$d_h = \begin{cases} d_{h-1} + t_{h-1,h}, & \text{if } i < h \leq j \\ t_{M,h}, & \text{if } h = i \text{ (in this case the} \\ & \quad \text{vehicle starts from } M) \end{cases}$$

Hence, $M(i,j)$ is simply given by $d_j + t_{j,M}$. Note that $M(i,j+1) = M(i,j) - t_{j,M} + t_{j,j+1} + t_{j+1,M}$. This means that all $M(i,j)$ can be computed in $O(nc)$ assuming $j \leq i+c-1$. Similarly, if batch $\{i,j\}$ starts indeed at time t, the contribution of job J_h to the objective function is given by:

$$f_h(t+d_h), \forall i \leq h \leq j$$

Clearly, $K(i,j,t)$ is given by $\sum_{h=i}^{j} f_h(t+d_h)$. Again, assuming that $f_j(\cdot)$ can be computed in constant time, note that $d_{j+1} = d_j + t_{j,j+1}$ and $K(i,j+1,t) = K(i,j,t) + f_{j+1}(d_{j+1})$. So, all values $K(i,j,t)$ can be computed in $O(ncT)$.

Once all values $M(i,j)$ and $K(i,j,t)$ are known, one can compute formula (17) for all feasible triples (i,j,t). Each such computation requires comparing nT values. Finally, $O(cT)$ values are compared to find the optimal solution. Since the feasible triples are $O(ncT)$, the computation of all values $F(i,j,t)$ clearly dominates the other phases, and the following result is proved.

Theorem 3.1. *Problem* $P1(\sum_j f_j(D_j))$ *can be solved in* $O(nc^2T^2)$.

4 CONCLUSION AND FUTURE DIRECTIONS

The problem treated in this paper takes place in a Supply Chain environment, where a manufacturer (modeled as a single machine) has to organize the delivery of the items (performed by a single capacitated vehicle). The sequence of production is given, and is supposed to be the same as the sequence of delivery. The problem is to form batches of jobs, so that the total delivery time is minimised. We prove that the problem is NP-hard for a capacity equal to 2, and a pseudopolynomial time dynamic programming algorithm is proposed.

Several research directions are possible. First of all, several polynomial cases can be proposed for this problem, and dynamic programming algorithms can certainly be proposed. Second, finding the complexity of the problem if the number of sites is smaller than the number of jobs. For the general case when the sequences have to be determined, heuristic algorithms can be proposed, as well as decomposition approaches.

REFERENCES

Agnetis, A., Aloulou, M. A., and Fu, L.-L. (2014). Coordination of production and interstage batch delivery with outsourced distribution. *European Journal of Operational Research*, 238(1):130 – 142.

Armstrong, R., Gao, S., and Lei, L. (2008). A zero-inventory production and distribution problem with a fixed customer sequence. *Annals of Operations Research*, 159(1):395–414.

Chang, Y.-C. and Lee, C.-Y. (2004). Machine scheduling with job delivery coordination. *European Journal of Operational Research*, 158(2):470 – 487. Methodological Foundations of Multi-Criteria Decision Making.

Chen, B. and Lee, C.-Y. (2008). Logistics scheduling with batching and transportation. *European Journal of Operational Research*, 189(3):871 – 876.

Chen, Z.-L. (2010). Integrated production and outbound distribution scheduling: Review and extensions. *Operations Research*, 58(1):130–148.

Fan, J., Lu, X., and Liu, P. (2015). Integrated scheduling of production and delivery on a single machine with availability constraint. *Theoretical Computer Science*, 562:581 – 589.

Garey, M. R., Tarjan, R. E., and Wilfong, G. T. (1988). One-processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research*, 13(2):330–348.

Lenté, C. and Kergosien, Y. (2014). Problème de livraison a séquence fixée. In *10ème conférence internationale de modélisation, optimisation et simulation (MOSIM'14), Nancy*.

Li, C.-L. and Ou, J. (2005). Machine scheduling with pickup and delivery. *Naval Research Logistics (NRL)*, 52(7):617–630.

Li, C.-L., Vairaktarakis, G., and Lee, C.-Y. (2005). Machine scheduling with deliveries to multiple customer locations. *European Journal of Operational Research*, 164(1):39 – 51.

Tsirimpas, P., Tatarakis, A., Minis, I., and Kyriakidis, E. (2008). Single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research*, 187(2):483 – 495.

Viergutz, C. and Knust, S. (2014). Integrated production and distribution scheduling with lifespan constraints. *Annals of Operations Research*, 213(1):293–318.

Wang, X. and Cheng, T. (2009). Production scheduling with supply and delivery considerations to minimize the makespan. *European Journal of Operational Research*, 194(3):743 – 752.